

FUSION OF IN-VEHICLE SENSOR DATA TO DEVELOP INTELLIGENT DRIVER TRAINING SYSTEM (IDTS)

Husnain, Malik

Andry, Rakotonirainy

Gregoire S., Larue

Center Of Accident Research And Road Safety – Queensland

Frederic, Maire

School Of Information Technology

Science And Technology Faculty

Queensland University of Technology

Australia

Paper Number 09–0048

ABSTRACT

The over represented number of novice drivers involved in crashes is alarming. Driver training is one of the interventions aimed at mitigating the number of crashes that involve young drivers. To our knowledge, Advanced Driver Assistance Systems (ADAS) have never been comprehensively used in designing an intelligent driver training system. Currently, there is a need to develop and evaluate ADAS that could assess driving competencies. The aim is to develop an unsupervised system called Intelligent Driver Training System (IDTS) that analyzes crash risks in a given driving situation. In order to design a comprehensive IDTS, data is collected from the Driver, Vehicle and Environment (DVE), synchronized and analyzed. The first implementation phase of this intelligent driver training system deals with synchronizing multiple variables acquired from DVE. ^{RT}Maps is used to collect and synchronize data like GPS, vehicle dynamics and driver head movement. After the data synchronization, maneuvers are segmented out as right turn, left turn and overtake. Each maneuver is composed of several individual tasks that are necessary to be performed in a sequential manner. This paper focuses on turn maneuvers. Some of the tasks required in the analysis of ‘turn’ maneuver are: detect the start and end of the turn, detect the indicator status change, check if the indicator was turned on within a safe distance and check the lane keeping during the turn maneuver. This paper proposes a fusion and analysis of heterogeneous data, mainly involved in driving, to determine the risk factor of particular maneuvers within the drive. It also explains the segmentation and risk analysis of the turn maneuver in a drive.

INTRODUCTION

Automobiles have greatly improved the transportation of goods and people around the globe. This factor in-return has enabled us to advance in many other areas. Crashes have been the most prominent danger associated with automobiles. These often result in serious injuries or loss of human life. Over 10 million people are injured yearly worldwide in road accidents. These include two to three million severely injured and 400,000 fatalities [1].

It is well known that drivers are at a greater risk during the early years of driving. About 95 per cent of all accidents are attributed to the human factor [2], whether it is driving too fast, lack of experience or simply misjudging a dangerous situation. Research indicates that young drivers are over represented in crashes because of a lack of experience, poor hazard perception, and a tendency to take risks [3,4]. Research suggests that the best learning environment for the inexperienced driver is the real road system under the supervision of an experienced driver or instructor [3,4].

Driver perception and learning of a particular driving hazard remains a key factor impacting road safety. In-order to comprehensively tackle road safety issues, a complete and integrated framework need to be developed that would include and examine all the parameters that influence driving (i.e. cues related to road, vehicle and driver). This requires the need for a system that can assess multiple maneuvers in a driving scenario as high risk or low risk based on the parameters acquired from DVE. This paper focuses on decomposing and analyzing turn maneuvers.

Figure 1 illustrates three sensors, namely FaceLab (eye tracking system), MobileEye (lane and obstacle detection system) and Vigil System (GPS and vehicle dynamics data logger) to gather data from the driver, environment and vehicle respectively. RTMaps is used to synchronize data from all the above mentioned sensors.

The primary driving tasks are divided into three broad categories: navigation and routing, guidance and maneuvers, and control [5].

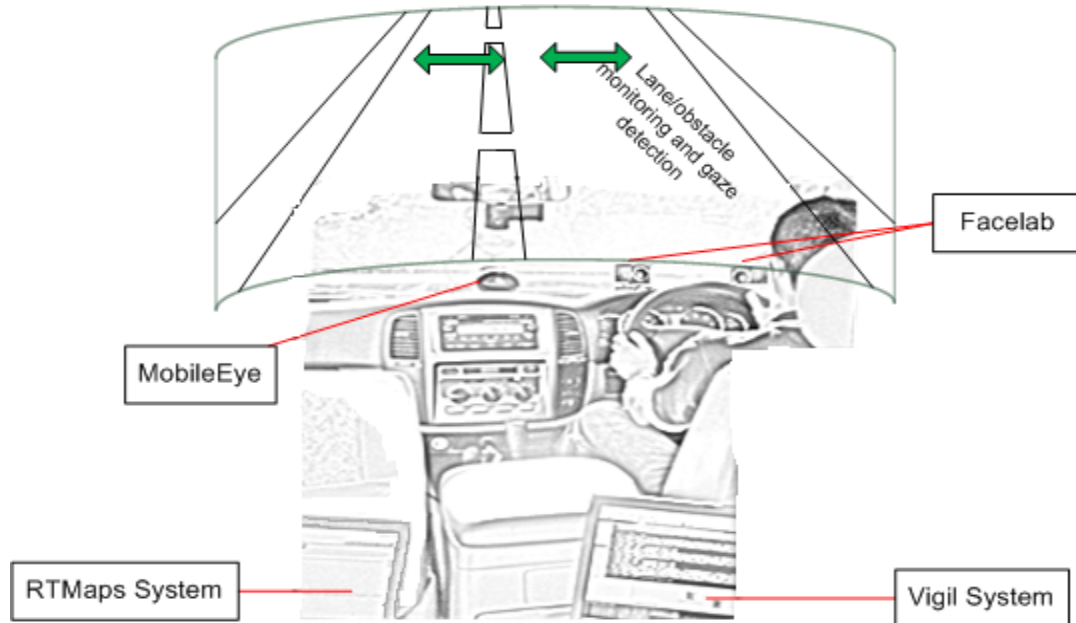


Figure 1. Multi-sensor system structure for Intelligent Driver Training System (IDTS)

The rest of the paper is organized as follows: the next section will briefly mention the related research work in the field of modeling an integrated driving scenario. The following section will comprehensively present our approach for developing the Intelligent Driver Training System (IDTS). This will be followed by mapping of the drive and future work section. Discussion and conclusion will be presented in the final section.

BACKGROUND RESEARCH

Assessing Primary Driving Tasks

It is well known that drivers are at a greater risk during the early years of driving. Researchers in [5] have defined the primary driving tasks as functions that are central to driving and without which moving a vehicle to a destination safely would not be possible.

Many intelligent systems have focused on warning the driver by predicting the trajectory of an oncoming obstacle [7], [6]. Only a few of these systems evaluate the overall driving situation and need to make the driver aware of relevant contextual knowledge extracted from sensors [8].

Execution of all these tasks is necessary for the driver in order to drive effectively.

Responding to critical events during driving requires timely response. A point reiterated in literature critical of driver training is that more in-depth analysis of the driving task and traffic situations is required. This analysis should take into account the cognitive skill aspect such as hazard and risk perception, decision-making, self-monitoring processes, learning styles, and risky attitudes to improve training [9, 7].

Sensors And Data Synchronization

Sensor fusion combines several sensor measurements in order to enhance the knowledge about the state of an object under observation. To increase the safety and efficiency for transportation systems, applications need to combine and comprehensively evaluate the data acquired from multiple sensors. Over the years, different type of sensors like radars, Global Positioning System (GPS), accelerometers, gyroscopic sensors and cameras have been used extensively in Advanced Driving Assistance Systems [6,7,8].

Various sensors can be used to perform obstacles detection: laser scanner, radar range, sonar range, vision (monocular or binocular). Similarly a number of sensors have been developed to measure vehicle dynamics as well. Researchers are now using in vehicle mounted sensors to measure different aspects of driving experience i.e. fatigue, monotony, body movements etc. Researches [15,16] have emphasized the usefulness of capturing driver gaze behaviour in creating a robust driving model.

Driving is a complex task. A single sensor alone is not enough to analyze such a task in a reliable manner. For example, GPS data has limited ability to describe or explain a driving situation. Furthermore a sensor can fail and produce erroneous data. In-order to model a complex driving scenario, multiple sensors data has to be merged to give a good representation of driving activity. Another hurdle in modeling a driving activity is that driving maneuvers can be performed with multiple styles. For example, indicator might not always be used just to signal lane change, or an overtake maneuver could involve a burst of speed but could also be performed by not accelerating hard at all [14]. Therefore it is necessary to view the multisensory data as a whole system to comprehensively model the driving activity.

A successful solution has to combine the benefits of multiple sensors such as GPS, radar, lidar and cameras. In order to obtain a precise synchronization, a sufficiently accurate global time for all sensors and fusion system is necessary. Therefore, to obtain a time consistent state for all sensors, the measurements have to be integrated in the order they were received. IDTS addresses these tasks by combining GPS, cameras and vehicle dynamics data using RTMaps.

METHODOLOGY

Architecture Of IDTS

To model a complex driving scenario in a comprehensive way, it is necessary to fuse several sensors data. Our test vehicle is equipped with vision systems, and sensors to monitor the vehicle dynamics as described in Figure 1.

Currently the test vehicle for this project includes the following sensors.

- Mobile Eye: It is a forward collision warning system that uses a single camera mounted on the windscreen of a vehicle. It also calculates variables such as distance from right/left lane and time to impact [10].
- FaceLab: It is a flexible and mobile tracking device that tracks head pose, eyelid movement

and gaze direction in real-time, under real-world conditions unobtrusively. This data can then further be used to monitor driver attentiveness, fatigue e.t.c [11].

- Vigil System: This visual-based management software program analyzes several areas of driving performance. Using GPS, accelerometers and cameras it measures speeds, accelerations, braking, cornering, following distances. The GPS input from this system is used to accurately view the vehicle's trajectory [12].
- RTMaps: It is the software that allows real time multiple data acquisition, data fusion and processing, at a high rate. The acquired data can also be stored for future replay. In this system, RTMaps is responsible for gathering data from the above mentioned systems (i.e. MobileEye, Facelab and vigil system), assigning a timestamp to it, synchronize it and storing the data. [13]

Sensor data fusion and the layered architecture of IDTS are shown in Figure 2. By fusing in multisensory data input, the precision and certainty of calculated estimates is increased e.g. the speed of the vehicle acquired from the odometer can be checked against the speed calculated from the GPS to remove any uncertainty. In this system's architecture, there is a bottom up stream of information acquired from multiple sensors.

As we can see in Figure 2 that the fusion layer is separate from the application layer (i.e. interpretation and assessment layer). This low coupled layered architecture is useful because the application layer does not require any interfacing with individual sensors. This scalable design helps in having multiple application layers while just having one sensor fusion layer (this is the only layer that has to have some knowledge of the sensor's characteristics).

As already mentioned, in this project the task of fusing sensory data input is handled by RTMaps [15]. It timestamps and synchronizes the sensor inputs from MobileEye, Facelab and VigilSystem during the drive. It then stores this drive data for future real-time replay. Processing of Facelab data is currently in progress.

The application layer (i.e. Interpretation + Assessment) described in Figure 2 handles the risk assessment of the maneuvers in the recorded drive. Currently, the interpretation layer detects the start, end and centroid of the turns from the drive (see 'Vehicle Turn Angle Estimation for a Turn Maneuver' section for more information). It then resolves the position where and if the indicator was turned on. It also removes the GPS

uncertainty by calculating the error variance (see ‘Estimation of GPS uncertainty’ section for more detail). After the classification of the turn, interpreted data is transferred to risk assessment layer where the distance between the start of the turn and start of the indicator is calculated. The distance of the vehicle from the right and left lane along with the speed is also considered to identify the risk involved.

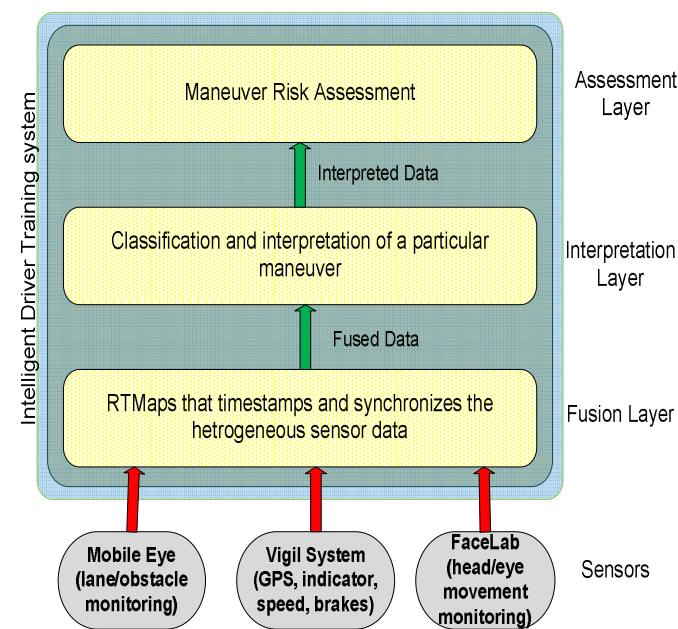


Figure 2: IDTS architecture and processing layers

Risk Assessment Criteria

There are a number of events that frequently occur during driving. A typical driving scenario would comprise of a certain set of driving events and patterns that are repeated over time.

Driver instructors typically assess a certain set of skills to assess drivers during a driver training session. They use various types of standard check lists to assess driving performance. For example, the analysis of a right-hand turn consists of observing a substantial number of sub events. The breakdown of this particular behaviour as stated in a sample driver training manuals is shown in Figure 3.

In the context of a driver training system, this list of driving events will be assessed automatically through the combined information gathered from the in-vehicle recording devices featuring multiple sensors and algorithms used to analyse video data.

Assessment such as the one described in Figure 3 will be used to develop a safe performance protocol model that can be used to assess automatically risk associated with a particular driving maneuver. In-order to achieve a less risky driving situation, a driver would have to perform these tasks properly in a sequential and timely manner. The model will be used as a formal framework to evaluate the perceptual and cognitive skills of the driver.

(Example of) DRIVER EDUCATION PERFORMANCE
<p>Right Turn Assessment</p> <ol style="list-style-type: none"> 1) Checks mirrors 2) Positions car properly in lane 3) Signals right 4) Reduces speed and keeps wheels straight 5) Checks traffic thoroughly, yielding to pedestrians 6) Starts turn when front wheels are opposite point where curb begins to curve 7) Uses proper steering when going into turn 8) Turns into proper lane 9) Straightens the wheels by using hand-over-hand, or methods maintaining secure control of steering 10) Adjusts speed to traffic flow <p>Source: Michigan Department of Education (1997, p35)</p> <p>Figure 3: Driver Education Performance</p>

Using such a system, an accurate measurement of the interaction between the driver, environment and the vehicle will be calculated. Table 1, identifies the sensors or technologies that output data to monitor each sub-event featured in a turn maneuver. For example, by using Facelab’s estimates of driver’s eye and head movements, tasks like check mirrors and check traffic are verified.

Table 1: Driving Subtasks and monitoring sensors/technologies	
Turn Sub-Events	Sensors/Technology
• Checks mirrors	FaceLab
• Positions car properly in lane	Mobile Eye
• Signals right	Mobile Eye
• Reduces speed and keeps wheel straight	Vigil System, MobileEye
• Checks traffic thoroughly	FaceLab
• Starts turn	Turn analysis algorithms
• Turns into proper lane	MobileEye
• Straightens wheel while maintaining secure control	Vigil System
• Adjusts speed to traffic flow	GPS , MobileEye

Vehicle Turn Angle Estimation For A Turn Maneuver

In-order to effectively model a turn maneuver, it is necessary to determine the complete demographics of a turn. IDTS calculates when the vehicle's turn started, when it finished and determines the centroid and the angle of the turn (i.e. was it a 90 degree turn or 45 degree turn e.t.c.).

Estimation of GPS Uncertainty – Evaluation and management of sensor uncertainty is important in a multisensory environment. GPS uncertainty has to be measured to accurately map the trajectory of test vehicle.

GPS provides the coordinates of a location with certain accuracy depending on its quality. When mapping the trajectory of a moving vehicle, it is important to be able to detect that given two GPS points, whether the second consecutive GPS coordinates represents a new position of the vehicle. Such an issue can be handled through the computation of the experimental variability of the GPS equipment used.

The GPS was placed at a point and multiple recording (frequency 1Hz) were taken to obtain the numerical error variance of the GPS. Errors along the horizontal and vertical axis are independent and equal. So errors along one axis are normally distributed around zero with variance σ^2 . Equation 1, was used to compute the variance along each axis separately. Then the maximum value is set as the variance σ^2 .

$$\sigma = \sqrt{\frac{1}{N} * \sum_{i=1}^N [calcDist(gps(i), meanGPS)]^2} \quad (1).$$

where N is the number of GPS points. And *calcDist* is the function that calculates distance between two GPS points. Implementation of the function '*calcDist*' which is using Haversine [10] formula to calculate distance between two GPS points is given below.

$$\Delta lat = lat2 - lat1$$

$$\Delta long = long2 - long1$$

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \cdot \cos(lat2) \cdot \sin^2\left(\frac{\Delta long}{2}\right)$$

$$c = 2 \cdot a \tan^2\left(\sqrt{a}, \sqrt{1-a}\right)$$

$$dist = R \cdot c$$

where *lat1* and *lat2* are the latitudes for first and second GPS points respectively. *long1* and *long2* are the longitudes for the first and second GPS points respectively. *R* is the radius of the earth and *dist* is the calculated distance between the two GPS points.

Errors being normally distributed means 95% of the coordinates obtained by the GPS are distributed within two standard deviations σ around the true position (see Figure 4) [18]. Given a GPS set of coordinates, a 95% confidence interval for the true location can be obtained from the variance as follows:

$$calcDist(GPS.coordinates, TRUE.location) \leq 2\sigma \quad (2).$$

See implementation of '*calcDist*'

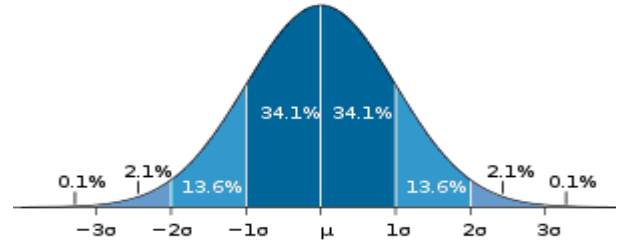


Figure 4. Gaussian distribution density function with mean μ and the variance σ

Numerically, we obtain $\sigma^2 = 0.0272m^2$, which corresponds to a true location inside a circle of radius 32.9cm. In other words two consecutive GPS points closer than 32.9cm cannot be considered as different.

Another issue related to the variability of GPS coordinates is that the direction of the moving vehicle given two consecutive GPS points can be insufficiently accurate for trajectory estimation (particularly in order to determine whether the vehicle is turning or not). In the worst case scenario the inaccuracy on the location estimation can lead to a difference in direction estimation by an angle θ as shown in Figure 5. Let A and B be the two absolute consecutive locations for a moving vehicle. In the worst case scenario, \tilde{A} and \tilde{B} represent the points obtained after the GPS error variance (of 2σ) in A and B respectively. So we can say that inaccuracy of vehicle heading estimation is θ .

For further analysis we require this angle to be small, and we take $\theta = 10^\circ$ as a threshold.

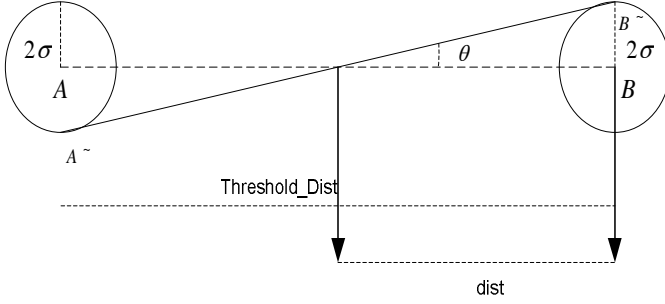


Figure 5. Calculation of Vehicle heading estimation θ

The threshold distance $threshold_dist$ is derived as follows:

$$threshold_dist = 2 * dist \quad (3).$$

From Figure 5, we can calculate that the distance (dist) is:

$$dist = \frac{2\sigma}{\tan\theta} \quad (4).$$

Using Equation 3 and 4, we get:

$$threshold_dist = \frac{4\sigma}{\tan\theta} \quad (5).$$

Numerically, we get $threshold_dist = 3.7m$. It means, two consecutive GPS points will be considered the same (i.e. the vehicle is stopped) if their distance is smaller than 3.7m. This threshold distance is useful in calculating the turn angle, as discussed in the following section.

Turn Detection Algorithm

Once the GPS error variance has been calculated, we compute an angle for every GPS point, in-order to determine the vehicle's turn angle. The algorithm for computing an angle for each GPS point (except for the first and last GPS point) is given below:

For all GPS points

Initialize t as 2

Store three points in array Y for time T-1, T and T+1

*Compute the **distance** between GPS points at time T-1 and T+1*

*While **distance** is less than threshold distance*

Add GPS points in Y array for time T-t and T+t, if only points exist for time T-t and T+t

Increase t by 1

*Compute **distance** between GPS points at time T-t and T+t*

*If **distance** is greater than or equal to threshold distance*

*Compute the **tangent angle** θ for GPS point at time T given T-t and T+t*

With the above mentioned algorithm, every GPS point will have a tangent angle based on the points before and after it.

Once the angle for every GPS point is calculated, the derivative of the angle with respect to the distance travelled ($\frac{\Delta\alpha}{\Delta s}$) is computed. $\Delta\alpha$ is the change in angle and Δs is change in distance. This derivative is useful in eliminating those GPS points during which the car didn't move a specified threshold distance. The method for computing the derivative $\frac{\Delta\alpha}{\Delta s}$ at each GPS point is very

similar to the method for computing angle θ for every GPS point. For the algorithm below, assume every GPS point now has an assigned angle as well (calculated using the above mentioned algorithm).

For all GPS points

Initialize t as 2

Store three GPS points in array Y for time T-1, T and T+1

*Compute the **distance** between GPS points at time T-1 and T+1*

*While **distance** is less than threshold distance*

Add GPS points in Y array for time T-t and T+t, if only GPS points exist for time T-t and T+t

Increase t by 1

*Compute **distance** between GPS points at time T-t and T+t*

*If **distance** is greater than threshold distance*

Compute the angle difference $\Delta\alpha$ between GPS points at time T-t and T+t

Compute the distance Δs between GPS points at time T-t and T+t

*Compute the **derivative** $\frac{\Delta\alpha}{\Delta s}$ for the GPS point T*

Figure 6 presents the vehicle trajectory in blue, while the red line represents the derivative $\frac{\Delta\alpha}{\Delta s}$ for the respective

GPS points. Based on these derivative values, the start, peak and end of the turn are segmented out. The start and end of the turn are crucial in finding out the centroid of the turn. This centroid is then used to calculate the 'safe' distance to switch on the indicator.

Centroid Calculation For The Turn

As already mentioned above the centroid calculation of the turns would be useful in identifying the ‘safe’ distance at which the driver switches the indicator before the turn. Usually, the exact start and exact end of the turn is debatable i.e. where do we decide that the car started to turn (e.g. when the driver started to turn the steering or when the car turned some significant angle). Therefore,

after the derivatives $\frac{\Delta\alpha}{\Delta s}$ for the whole drive have been

calculated, the turn is segmented out based on the start and end turn using heuristics. Once the turn has been segmented out from the drive, its centroid is calculated. Even if the exact start of the turn is ambiguous, the centroid of the turn would be always accurate.

The centroid of an area is similar to the center of mass of a body [19]. The centroid of the turn is calculated between the start and end of the turn (i.e. the turn area) using

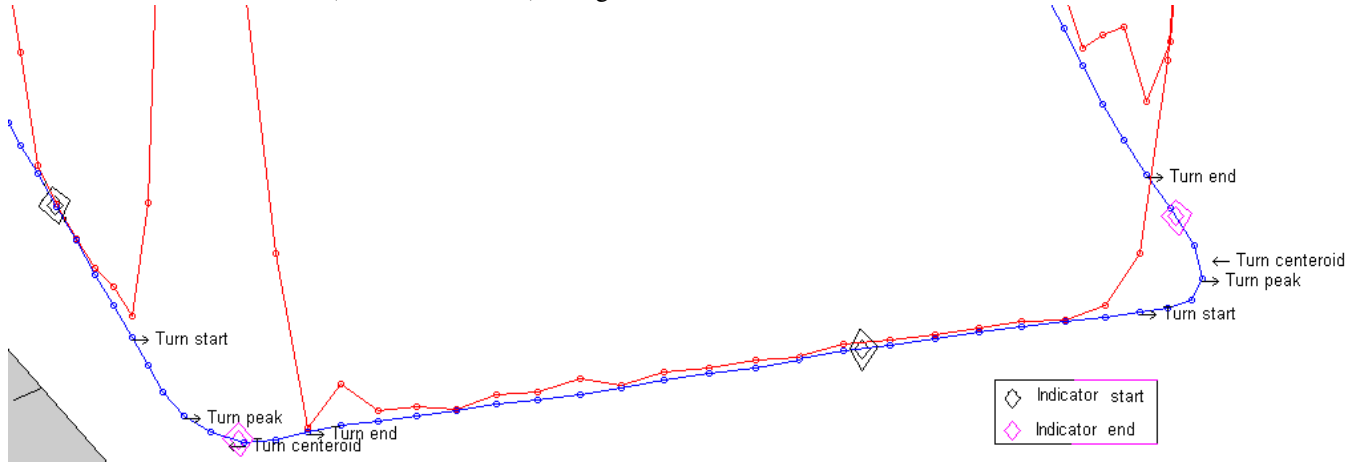


Figure 6. DRIVE1 – Representation of vehicle's GPS trajectory with two left turns (in blue) and the derivative values $\frac{\Delta\alpha}{\Delta s}$ plot for the corresponding GPS points (in red)

Figure 7, presents the turns involved in DRIVE 2. It consisted of three turns, first was a left turn followed by a right and finally a left turn. The derivate $\frac{\Delta\alpha}{\Delta s}$ values are plotted along Y axis and the number of GPS points around X axis. It is evident

from the graph that using derivatives, the exact nature of the turn can be deduced e.g. whether it was a left or a right turn (based on the sign of derivative). This data can also be used to compute the vehicle turn angle. From the graph, we can see that the turns can be segmented out from the rest of the drive using heuristics based on the derivative values. All this information coupled with indicator, gaze and lane keeping data effectively model the turn scenario.

derivative $\frac{\Delta\alpha}{\Delta s}$ as a weight function A_N (see equation 6).

Calculating the centroid involves only the geometrical shape of the area. So the area is divided into multiple rectangles and using Equation 6 below, the centroid of the area is calculated.

$$C = \frac{\sum A_N C_N}{\sum A_N} \quad (6).$$

where C_N is the index of the N^{th} GPS point in the turn area and C is the centroid of the turn.

Figure 6, illustrates the turn's centroid for both turns in DRIVE 1. Other information, like indicator start, indicator end and turn start/end are helpful in accurately modeling these turns.

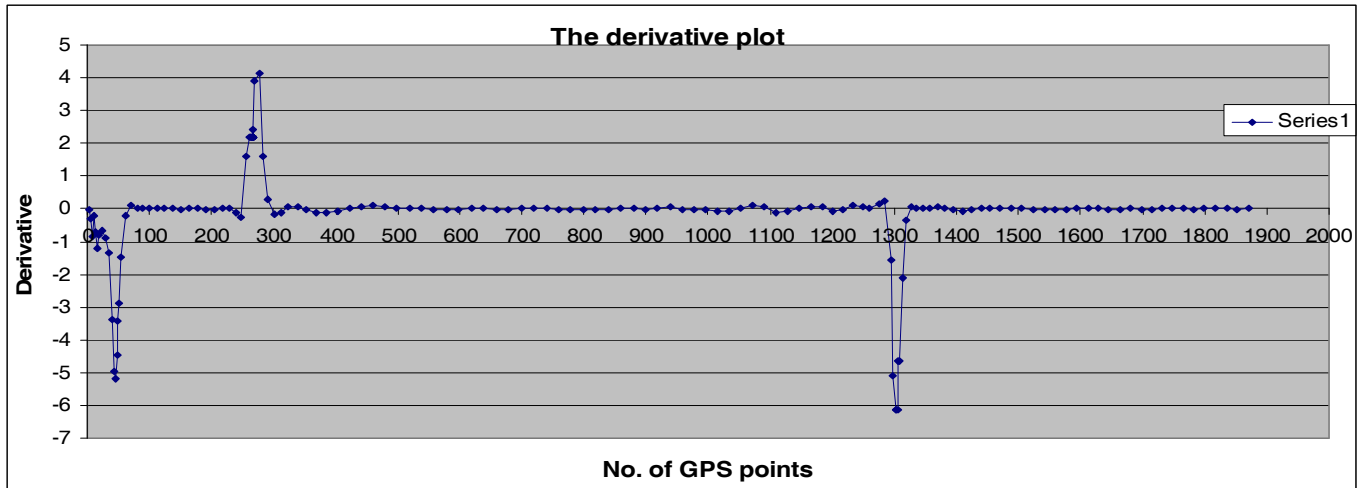


Figure 7. DRIVE 2 – Represents the derivative values $\frac{\Delta\alpha}{\Delta s}$ along Y axis and No. of GPS points on X axis.

MAPPING

Visualization of the drive is an integral part of this project. Since its end users are driver trainers, it is necessary that all drive data and risky situations are represented in a way that is easy to comprehend. Hence, it will be easy for the driver trainers to explain some specific situation to the driver.

Figure 8, presents an example of the vehicle trajectory and drive data for Drive 1. This interactive user interface would help drivers and their trainers to assess certain maneuvers in a drive by combining the multidimensional data acquired from DVE. By combining the numerical information from the graph in Figure 6, this interactive map (Figure 8) is able to show distance between indicator switch on and the turn start/turn centroid. It is also able to show if during a maneuver, driver followed the lane keeping procedure.



Figure 8. DRIVE1 – Representation on map where indicator on (yellow) and indicator off (pink) were performed.

FUTURE RESEARCH

In-order to effectively model a turn maneuver, driver's gaze direction should be tracked as well. Different eye tracking systems together with head tracking algorithm are suggested based on near infra-red or visible light using different hardware architectures.

These systems, by calculating the gaze and head direction in 3D allow calculating the coordinates where the gaze intersects with the world (a virtual plane in-front of the driver). Using perspective projection techniques, we plan to calculate the approximate depth of a drivers' gaze in a real world. Furthermore, this gaze information would be presented on an interactive map. This approximate depth calculation would be very helpful in determining the difference of gaze pattern in experienced and novice drivers.

Along with this, it is necessary to comprehensively model all tasks required for a less risky turn. Further work would be required to model other maneuvers like overtake, roundabout e.t.c.

CONCLUSION

This paper presented a framework for analyzing a turn maneuver. The prototype (IDTS) currently, integrates information related to vehicle dynamics and road

information. Next step is to model driver's gaze data and integrate it in this turn maneuver. The information gathered from DVE will help to contextualize, observe and better assess a range of driving maneuvers. This prototype is the building block to evaluate driver's competency. It acts as a assisting tool for the driver trainers.

Eventually both drivers and driver trainers would be able to assess the drive using IDTS. As already mentioned, a major percentage of road crashes are attributable to driving error. Thus, driver training remains an important road safety intervention to improve driving performance and abilities, particularly amongst young people.

REFERENCES

- [1] National Transportation Safety Board, Special Investigation Report - Highway Vehicle- and Infrastructure-based Technology For the Prevention of Rear-end Collisions. NTSB Number SIR-01/01, May 2001
- [2] "BMW Driver Training Gears Up for Safer Australian Drivers", http://www.autoweb.com.au/cms/A_108783/title_BMW-Driver-Training-Gears-Up-for-Safer-Australian-Drivers/newsarticle.html
- [3] Mayhew, D. R. & Simpson, H.M. (1996). "The Effectiveness and Role of Driver Education and training in a Graduated Licensing System". Ottawa, Ontario: Traffic Injury Research Foundation.
- [4] Engström, I., Gregersen, N.P., Hernetkoski, K., Keskinen, E. and Nyberg, A. (2003). Young novice driver education and training: Literature review. VTI-rapport 491A. Linköping.: Swedish National Road and Transport Research Institute.
- [5] "Task analysis of ATIS/CVO Functions, Development of human factor guidelines for advanced traveler information system and commercial vehicle operations", Publ. No. FHWA-RD-95-176.
- [6] Massimo, Alberto, and Alessandra, "Vision-based intelligent vehicles: State of the art and perspectives," Robotics and Autonomous Systems, vol. 32, pp. 1-16, 2000.
- [7] Massimo, Alberto, and Alessandra, "Vision-based intelligent vehicles: State of the art and perspectives," Robotics and Autonomous Systems, vol. 32, pp. 1-16, 2000.
- [8] T. M. Manubhai, G. Tarak, and M. Joel, "Looking-In and Looking- Out of a Vehicle: Computer-Vision-Based Enhanced Vehicle Safety", ITS IEEE Transactions, vol. 8, pp. 108-120, 2007.
- [9] E. Andreas, P. Jochen, G. Fredrik, and E. Jonas, "Toward Autonomous Collision Avoidance by Steering", ITS, IEEE Transactions on, vol. 8, pp. 84-94, 2007.
- [10] Erez Dagan, Ofer Mano, Gideon P. Stein, Amnon Shashua, "Forward Collision Warning with a Single Camera", Intelligent Vehicles Symposium, 2004 IEEE.
- [11] Facelab.<http://www.seeingmachines.com/facelab.htm>
- [12] Vigil Vanguard System, <http://www.vigilsystems.com/13.0.html>
- [13] RTMaps: a software for new technologies, real-time and multisensor: www.intempora.com/ENG/maps/index.php?nav1=maps&nav2concept
- [14] Arati Gerders, "Automatic maneuver recognition in the Automobile: the fusion of uncertain sensor values using Bayesian models", 3rd international workshop of intelligent transportation, WIT 2006.
- [15] D.D. Salvucci, "Inferring driver intent: A case study in Lane-Change Detection", 48th Annual meeting of human factors and ergonomics society, USA 2004.
- [16] K. Torkolla, S. Venkatesan, H. Liu, "Sensor sequence modeling for driving", 18th international FLAIRS conference , Florida, USA 2005.
- [17] Nico Kaempchen, Klaus Dietmayer, "Data Synchronization Strategies For Multi-Sensor Fusion", Proceedings of 10th World Congress on Intelligent Transport Systems (ITS 2003), Madrid, Spain, 2003
- [18] Gaussian Distribution, www.cs.princeton.edu/introcs/11gaussian/
- [19] Centroid detection, Centroid Of An Area, <http://www.efunda.com/math/areas/Centroid.cfm>
- [20] Haversine formula, Calculate distance between two Latitude/longitude points <http://www.movable-type.co.uk/scripts/latlong.html>